

Flutter & Dart Complete Guide

Prepared by Eng. Yousef Salah Kassem

DAFL 2
Firebase

Lecture 10



Table Of Content

- GITHUB
- Firebase Deployment



It is always wise to save your codes on GitHub because it acts like a cloud if anything is to happen to your system such as hard disk failure, or even a software failure. GitHub will act like an external storage device. GitHub is also a place to showcase your codes to get reviews from other developers. With that being said let's look at a simple way to upload your flutter projects on GitHub.



Steps on how to upload Flutter projects to GitHub

Step 1. Go on GitHub and create a brand new repository.

Step 2. Give your repository a name and make sure public is checked then click on Create a new repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * Repository name *

Tosin2289 / new-repo

✔ new-repo is available.

Great repository names are short and memorable. Need inspiration? How about [vigilant-fortnight](#) ?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file

- Note: Make sure you have Git already installed on your device. if you haven't head over to <https://git-scm.com/> to get that done



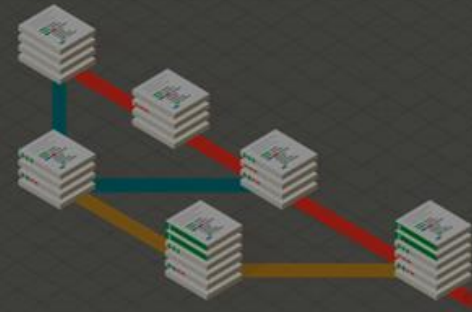


🔍 Type / to search entire site...



Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.



About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.



Mac GUIs



Tarballs



Windows Build



Source Code



Pro Git by Scott Chacon and Ben Straub is available to **read online for free**. Dead tree versions are available on [Amazon.com](#).

Documentation

Downloads

GUI Clients

Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to **read online for free**. Dead tree versions are available on [Amazon.com](#).

Downloads



macOS



Windows



Linux/Unix

Older releases are available and the **Git source repository** is on GitHub.



GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

View Logos →

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the **web interface**.

Step 4. Move back to your code editor(visual studio code or android studio) toggle over to the terminal and enter the following commands in your project folder.

```
git init
git add .
git commit -m "first commit"
git branch -M main
git remote add origin https://The link to your git repo
```

```
git push -u origin main
```



**Set up GitHub Copilot**
Use GitHub's AI pair programmer to autocomplete suggestions as you code.

**Invite collaborators**
Find people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

 Set up in Desktop **or** **HTTPS** **SSH** `https://github.com/Tosin2289/new-repo.git` 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

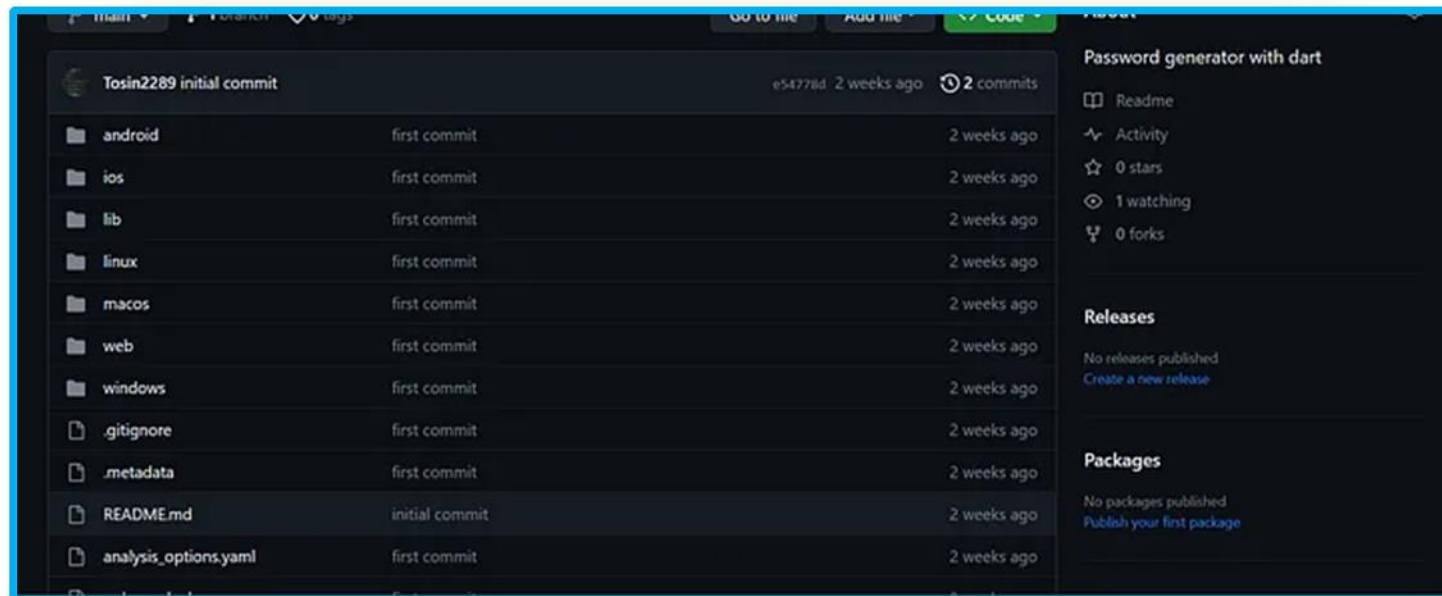
```
echo "# new-repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Tosin2289/new-repo.git
git push -u origin main
```



Chapter 10

GITHUB

After doing all this head over to your GitHub and refresh the page you should have something like this.



Firebase Deployment

Firestore Deployment

Firestore App Distribution makes distributing your apps to trusted testers painless. By getting your apps onto testers' devices quickly, you can get feedback early and often.

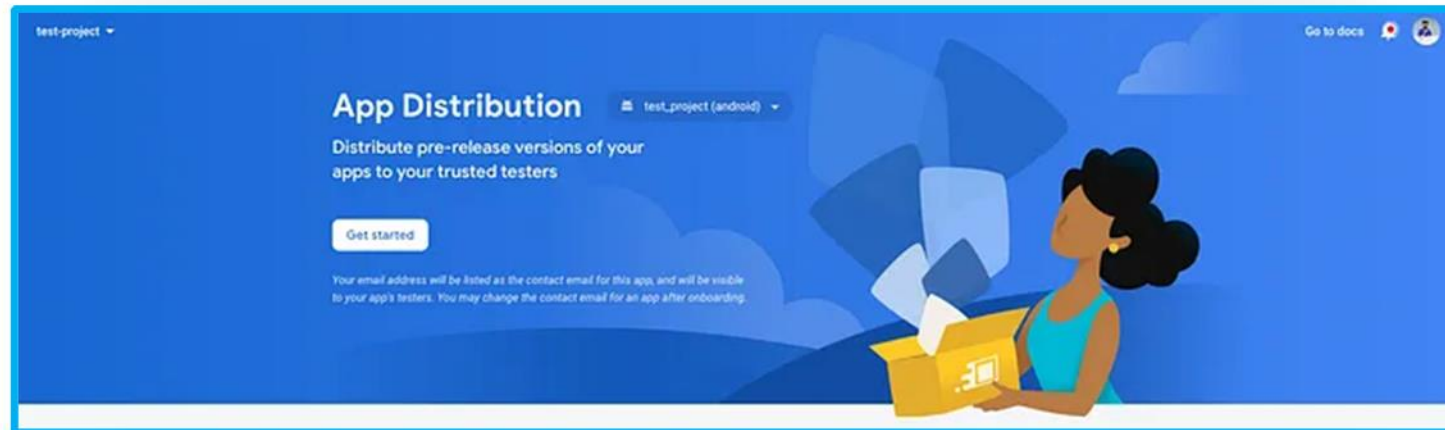


Chapter 10

Firebase Deployment

Enable App Distribution from Firebase Console

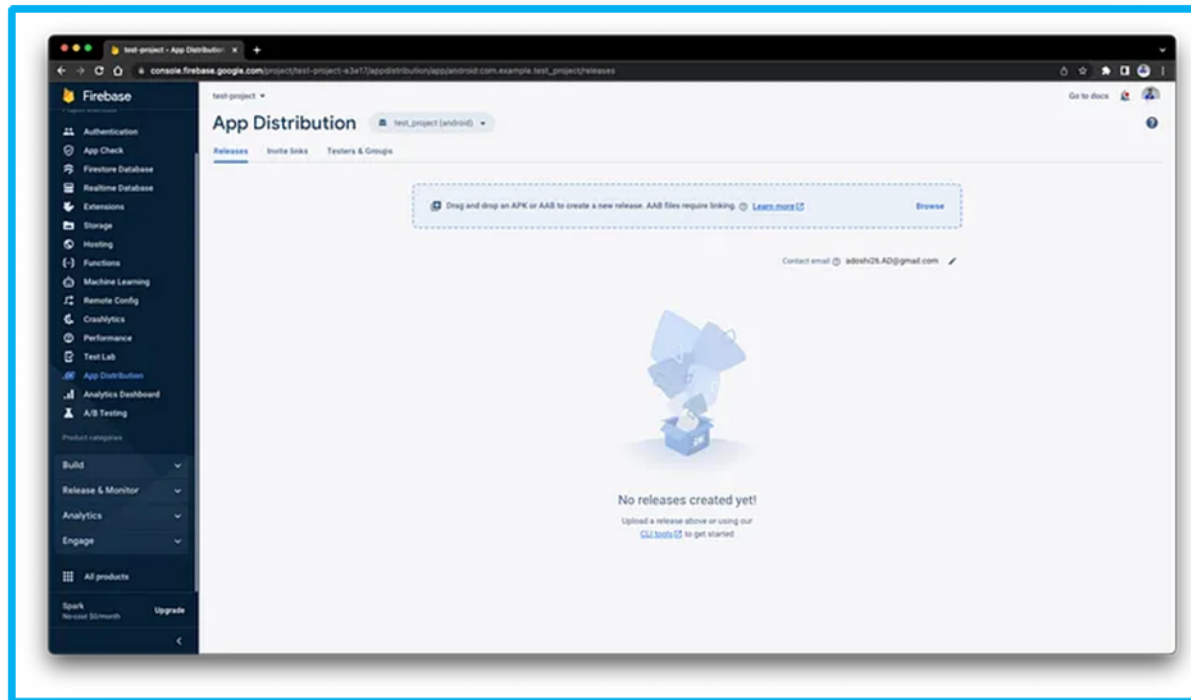
The next step is to go to [Firebase Console](#) into your project and enable App Distribution. You can find App Distribution in the right panel under **Release & Monitor**.



Chapter 10

Firebase Deployment

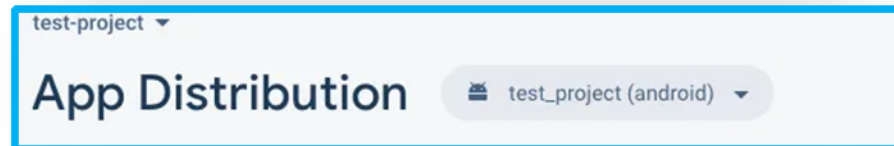
Once you click on **Get started**, you will reach the App Distribution home page which looks something like the following:



Firestore Deployment

Step 2: Select Android or iOS

On top of the screen, there will be a drop-down menu where you can choose whether you want to upload an Android build (APK or AAB)



Firebase Deployment

Step 3: Add your build to Firebase App Distribution

Before deploying to Firebase, build your Flutter app for release. Open a terminal and run:

```
flutter build apk - release # For Android
```

```
flutter build ipa - release # For IOS
```



Firebase Deployment

Step 3: Add your build to Firebase App Distribution

Before deploying to Firebase, build your Flutter app for release. Open a terminal and run:

```
flutter build apk - release # For Android
```

```
flutter build ipa - release # For IOS
```

Here comes the main step! Now in the **Release** tab, drag and drop or browse the build (APK/IPA) into the Firebase App Distribution.

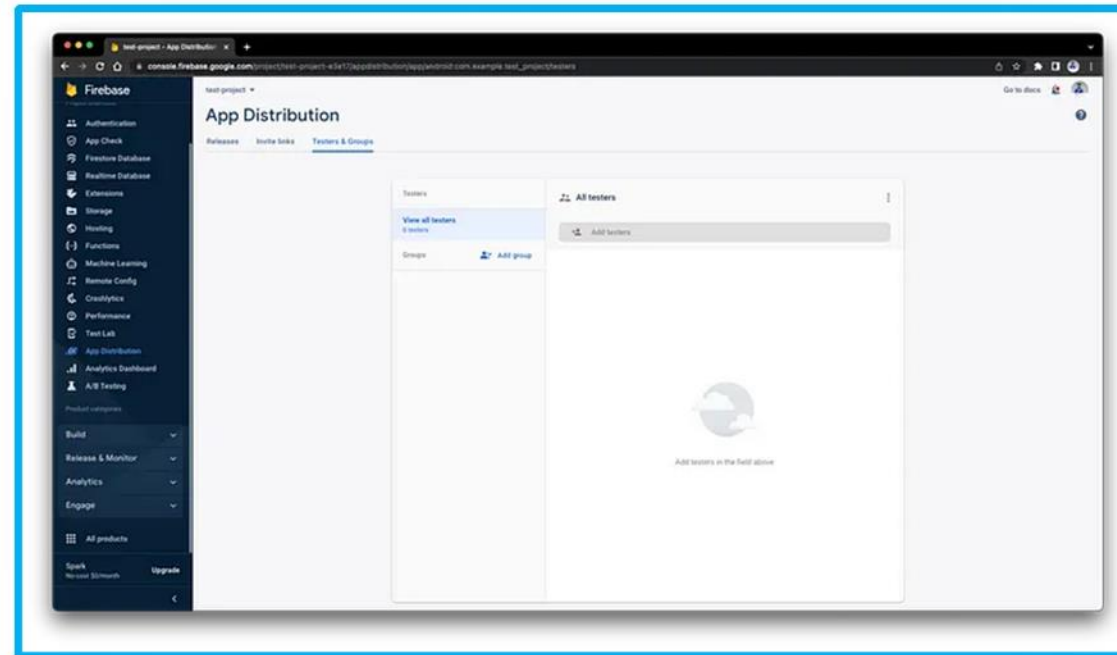


Chapter 10

Firestore Deployment

Step 4: Manage Testers

Once the build is successfully uploaded, go to the **Testers & Groups** tab.



Firestore Deployment

Here, you can add a group of testers, for example, **internal testers** can be one of the group, **beta testers, etc.** You can also add individual testers in this section itself. The build that you upload, will be distributed to these testers. You can check the [manage testers](#) docs for more on creating tester groups.



Firestore Deployment

Step 5: Distribute!

The last step is to click on **Distribute**, and it will be available to all your testers. That's it. Done and dusted! Your app is sent to your testers.



Live Coding

What is Next ?



Thank You